

Multi-domain large deformations by monolithic approach for massive parallel computing

F. EL HADDAD ^{a,*}, H. DIGONNET ^a, T. COUPEZ ^a

^a Cemef, Mines Paristech
1 Rue Claude Daunesse, 06904 Sophia Antipolis, FRANCE

*fadi.el_haddad@mines-paristech.fr

Abstract. *A monolithic approach is presented to solve multi-domain large deformations problems. It is based on an eulerian formulation using a fixed yet anisotropic adaptative mesh. Within this context, a level set method is used to capture the interface evolution of different domains. In terms of parallel computing, it is expected that the technique of single mesh and single mechanical solver developed under CimLib allows to obtain easily a very good scalability up to several hundred processors.*

Keywords: monolithic approach; level set; anisotropic adaptation; parallel computing.

1 INTRODUCTION

Multi-domain eulerian formulations are used to solve a broad range of problems in CFD and FSI. Indeed such formulations have shown a strong ability handling large deformations and generation of new surfaces. Therefore, this paper will extend the theory towards problems in solids mechanics specially forging processes.

Most of the numerical approaches solving such problems have proposed a Lagrangian approach or arbitrary eulerian Lagrangian approach [2, 6]. However, important problems arise when interactions between several domains occur. In a multi-domain Lagrangian approach each sub-domain have its own independent velocity field therefore a contact algorithm is required [8]. In contrast, the eulerian formulation prevents two domains from occupying the same space due to the single continuous velocity field. An eulerian approach has been developed in [3] using an explicit interface tracking method. Thus, the aim of this work is to present a “fully” eulerian approach using an implicit interface capturing. In fact, the description of different domain interfaces is determined implicitly using a level set method [7]. Within this context the mesh does not represent anymore the interacted domains, it serves only to solve a global set of equations defined on a whole domain englobing all sub-domains. In such an approach, all difficulties brought by the parallelization of contact algorithms are no more needed. In this case, the efficiency of the parallelism depends only on the partition of the used single mesh.

2 MONOLITHIC APPROACH

2.1 Governing Equations

In this section, a brief description of the monolithic approach will be presented. It consists on solving one global set of equations defined on the whole computing domain. Let $\Omega = \cup \Omega_i \subset \mathbb{R}^d$ ($d = 2, 3$) be the computing domain where Ω_i represents the different interacted domains. The mechanical problem is governed by the mass conservation equation and the momentum equation, in which the inertia term is neglected and defined as following:

$$\begin{aligned}\nabla \cdot \sigma &= \rho g \\ \nabla \cdot v &= 0\end{aligned}\tag{1}$$

where σ is the stress tensor, ρ is the density, g is the gravity and v is the velocity.

The stress tensor σ is expressed as $2\eta_i\varepsilon(v) - pI$, where η_i , ρ_i represent respectively the consistency and the density of each sub-domain Ω_i , $\varepsilon(v)$ represents the strain rate tensor, p the pression and I the identity matrix .

2.2 Level Set

A standard level set method is based on the advection of the signed distance function Φ :

$$\begin{cases} \Phi(\vec{x}) = \pm d(\vec{x}, \Gamma) \quad \forall \vec{x} = (x_1, \dots, x_d) \in \Omega \\ \Gamma = \{\vec{x} \in \Omega / \Phi(\vec{x}) = 0\} \end{cases} \quad (2)$$

Since the complete level set advection is not necessary and the only useful information is located in the vicinity of the interface, using a hyperbolic tangent filter proves to be handy $\hat{\Phi} = \varepsilon \tanh \frac{\Phi}{\varepsilon}$; where ε represents the thickness of the interface. In order to retain Φ as a distance function a reinitialization step is needed. A conservative level set method is introduced in [9] capable of advecting and reinitializing the level set in the same time.

2.3 Mixture laws

After defining all sub-domains geometrically with a distance function, the physical characteristics of each sub-domain are determined by mixture laws. To represent density and viscosity discontinuities over the interface a Heaviside function is needed. In computations, to achieve numerical robustness, a smoothed Heaviside is often used:

$$H_\varepsilon(\Phi) = \begin{cases} 0 & \text{if } \Phi < -\varepsilon \\ \frac{1}{2} \left(1 + \frac{\Phi}{\varepsilon} \right) & \text{if } |\Phi| < \varepsilon \\ 1 & \text{if } \Phi > \varepsilon \end{cases} \quad (3)$$

Now, the consistency is defined on the whole domain in terms of the Heaviside and the level set functions:

$$\eta(H_\varepsilon) = \sum_{i=0}^n \eta_i \times H_\varepsilon(\hat{\Phi}_i) \quad (4)$$

2.4 Anisotropic mesh adaptation

Coupled with the Level set method, the anisotropic mesh adaptation is applied to adapt the interface. In the interface vicinity, the mesh becomes locally refined, the elements stretched in a precise direction. A metric tensor is essential to control the generation of the anisotropic mesh. It allows the mesh size to be imposed in the direction of the distance function gradient. The author of [11] proposes a new procedure to define an anisotropic mesh. The idea is to build the metric directly at each node of the mesh such that the lengths of connected edges to this node are close to unity.

$$\|x\|_M = {}^t x M x = 1$$

Further details of the metric calculation and the error estimator used to control the generation of the anisotropic mesh are available in [1, 10, 11] .

3 PARALLEL COMPUTING

The parallel computing has proved its importance in reducing the computational time, it is necessary to simulate industrial cases where the mesh contains a very great number of elements. However, communication between processors and parallelization of some algorithms may lead to loss of efficiency. The main criterion for judging the parallelization of an algorithm is the ‘‘Speed up’’.

Speed up refers to how much a parallel algorithm is faster than a sequential one; it is defined by the following formula:

$$S_p = \frac{T_1}{T_p} \quad (5)$$

where p is the number of used cores, T_1 represents the execution time of the sequential algorithm and T_p the execution time of the parallel algorithm using p cores. Theoretically, the optimal Speed up is obtained when $S_p = p$, i.e. when executing an algorithm using p cores, the parallel computing time should be equal to the sequential time divided by p .

4 NUMERICAL RESULTS

4.1 Multi-domain

The used computational domain is sized to $100\text{ mm} \times 100\text{ mm} \times 51\text{ mm}$. The deformable geometry contains ten equally sized parallelepiped ($10\text{ mm} \times 10\text{ mm} \times 5\text{ mm}$). The consistency of each deformable body is equal to 352 MPa.s . The imposed velocity in the upper tool is equal to 10 mm/s while the lower tool remains fixed. Each parallelepiped is represented by a level set function. The mesh contains 147 501 nodes and 802 340 elements. Figure 1 shows the initial disposition as well as a top view of the bodies deformed 80% of the geometry's initial height.

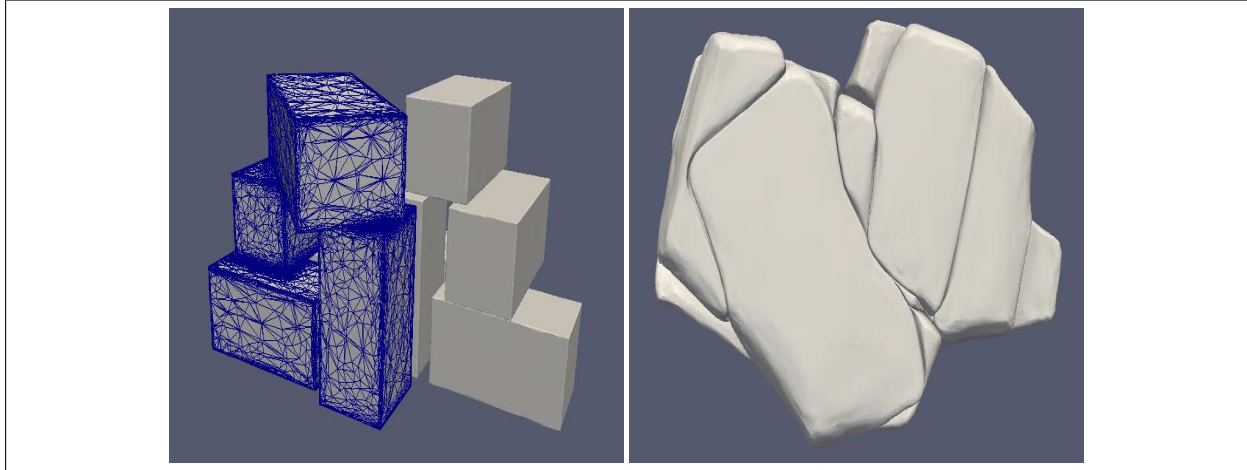


Figure 1: On the left: the initial geometry of a multi-domain formed by ten deformable bodies. On the right: top view of the final geometry after being deformed 80% of the geometry's initial height.

4.2 Parallel performance

The purpose of the following test case is to prove the effectiveness of parallelization used in "CimLib" [4]. A study was carried out in order to show the capacity of the "CimLib" library to manage massively parallel computations [5]. Using an isotropic mesh, containing 491 055 nodes 2 879 257 elements and a mesh size equal to 0.625 mm. The forging process was stopped when the forging reached a deformation of 10% of its initial height. Using "AMD Opteron 6 134" processors, the simulations have been launched locally on the nodes. A computing node is formed by two processors containing 8 cores each.

The "bind to core" option available in version 1.4.3 MPI, is used to attach a process to a core throughout the program. This option is essential for good performance.

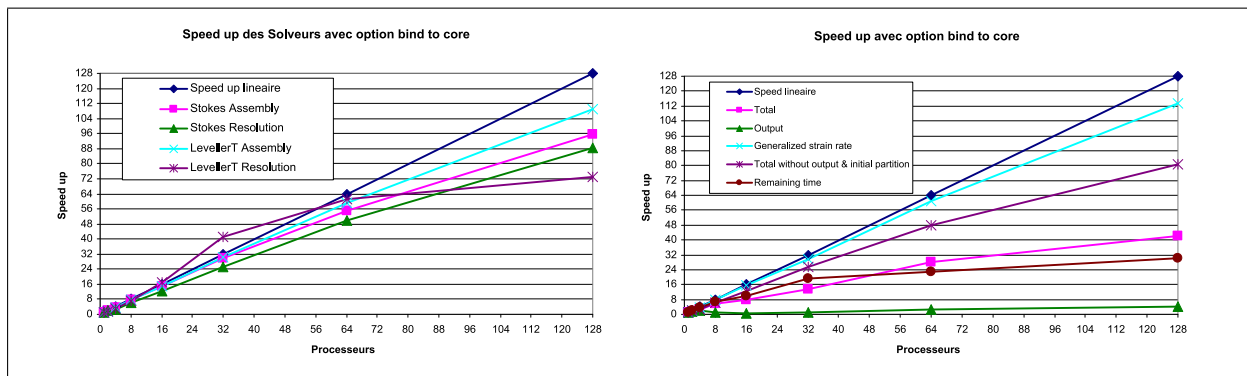


Figure 2: Speed up obtained in a flat bunch simulation; containing 491 005 nodes

Figure 2 shows the "Speed Up" obtained as a function of the number of cores. At the level of assembling and resolution, a slight loss of efficiency is present between 1 and 8 cores due to partial use of CPU for a small number

of core. However, beyond 8 cores acceleration is very close to the ideal. In addition, by subtracting the time taken to write the results and accomplish the initial partitioning of the mesh, the overall speed up is very close to the resolution of Stokes.

5 CONCLUSION

This paper presents an Eulerian monolithic approach. It allows writing the problem of large deformation using one system of equations defined on the whole domain. Different tools are used to identify precisely the position and the characteristics of each sub-domain in the work domain, such as distance functions and mixture laws. Better accuracy is obtained by using an anisotropic adapted mesh. Results have shown the capacity of this approach to simulate multi-domain deformations problems without losing its efficiency in term of parallel computing. The overall Speed up is almost the same for the resolution of the mechanical solver. As a perspective, the ability of the monolithic eulerian approach to describe different types of contact between several bodies will be studied. For example, the mixture law presented in this paper, allows to establish a bilateral sticking contact between bodies. Numerically, it leads to a viscosity of the same order at the material-tool interface, making it appear as friction between the tools and the deformable body. Other types of contact can be obtained by introducing a quadratic mixture law and substituting the scalar η by a tensor $\bar{\eta}$.

REFERENCES

- [1] C. Gruau, *3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric*. Computer Methods in applied Mechanics and Engineering, 194(48):4951-4976, 2005.
- [2] G. Surdon, J-L Chenot *Finite element calculations of three-dimensional hot forging*. International Conference on Numerical Methods in Industrial Forming Processes NUMIFORM'86, Mattisson K et al (eds). A.Balkema: Rotterdam, 1986; 287-292.
- [3] D. Benson, *A mixture theory for contact in multi-material Eulerian formulations*. Computer Methods in applied Mechanics and Engineering, 140:59-86, 1997.
- [4] H. Dignonnet, T Coupez *Object-oriented programming for 'fast-and-easy' developpement of parallel applications in forming processes simulation*. In second MIT Conference on Computational Fluid and Solid Mechanics, Massachusset Institute, Bathe KJ (ed.). Elsevier: Amesterdam, 2003; 1922-1924
- [5] H. Dignonnet, *Repartitionnement dynamique, mailleur parallèle et leurs applications à la simulation numérique en mise en forme des matériaux*. Thèse Ecole Des Mines De Paris, 2001.
- [6] J. Barboza, *Traitement du contact entre corps déformables et calcul parallèle pour la simulation 3D du forgeage multicorps*. Thèse Ecole Des Mines De Paris, 2004.
- [7] J. Bruchon, H. Dignonnet, T. Coupez *Using a signed distance function for the simulation of metal forming processes: Formulation of the contact condition and mesh adaptation. From a Lagrangian approach to an Eulerian approach*, Int. J. Numer. Meth. Engng, Wiley, 78:980-1008, 2008.
- [8] L. Fourment, JL Chenot, K Mocellin. *Numerical formulations and algorithms for solving contact problems in metal forming simulation*. International Journal for Numerical Methods in Engineering 1999; 46(9):1435-1462.
- [9] L. Ville, L.Silva, T. Coupez. *Convected Level set method for the numerical simulation of fluid buckling*. Int. J. Numer. Meth. Fluids, 2010.
- [10] T. Coupez. *Génération de maillage et adaptation de mailage par optimisation locale*, Revue européenne des éléments finis, 9(4)(2000) 403-423.
- [11] T. Coupez. *Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing*, Journal of Computational Physics, elsevier, 2391-2405, 2011.